## 解密openGauss DB4AI框架的内部机理

1. openGauss AI框架的特点

DB4AI这个方向中，数据库通过集成AI能力，在用户进行AI计算时就可以避免数据搬运的问题。不同于其他的DB4AI框架，本次openGauss开源的原生框架是通过添加AI算子的方式完成数据库中的AI计算。

那么除了避免了数据搬运所带来的问题这个普遍优势，openGauss的AI框架还具有一下的优势和特点：

1）极低的学习门槛

当前最主流的计算框架：Tensorflow、pytorch、keras等大多依托于python语言作为构建的脚本语言，虽然python已经足够的简单易学但还是需要一定的学习成本的。而目当前的框架，设计提供了CREATE MODEL和PREDICT BY两种语法用于完成AI的训练和推断任务。该语法相比较python更加趋近于自然语言，符合人们的用语直觉。

|  |
| --- |
| CREATE MODEL point\_kmeans USING kmeans FEATURES position FROM kmeans\_2d WITH num\_centroids=3;  SELECT id, PREDICT BY point\_kmeans (FEATURES position) as pos FROM (select \* from kmeans\_2d\_test limit 10); |

2）极简的数据版本管理

本次DB4AI特性中还添加了snapshot功能。数据库通过快照的形式将数据集中的数据固定在某个时刻，同样也支持保存经过处理过滤的数据。功能分为全量保存和增量保存，其中因为增量保存每次仅存储数据变化，快照的空间占用大大的降低了。用户可以直接通过不同版本名称的快照直接获取相对应的数据。

3）极优的性能体验

相比于目前很多的AIinDB项目，openGauss的特性通过添加AI算子的方式将模型计算内置到数据库中了。以算法训练为例，其中的数据的读取、模型的计算更新和最终的模型保存将在数据库的执行器中完成。这种方式将更加充分地利用和释放数据库的计算能力。深入内核的技术路线使得我们的特性在计算速度上将优于其他更高层级调用的方法。

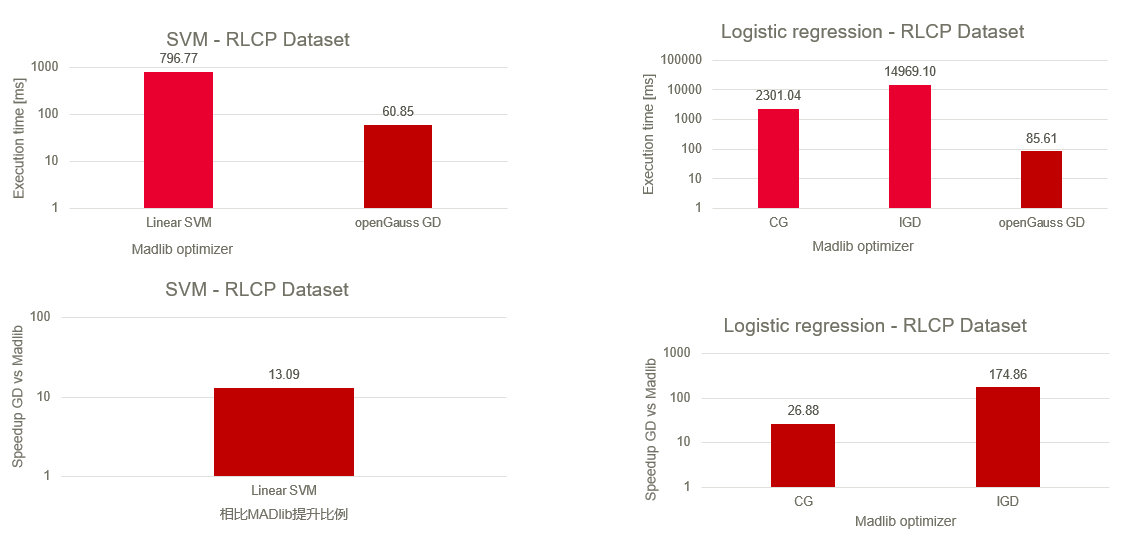
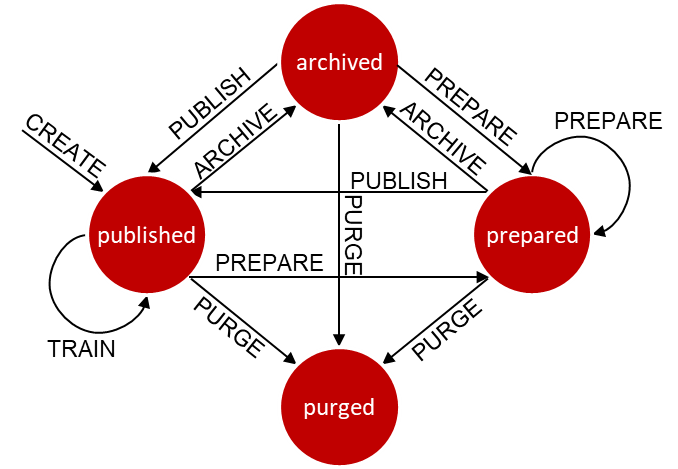


图1.与MADlib性能对比

1. 技术原理与优势
   1. DB4AI-Snapshot



首先DB4AI.snapshot特性需要用户通过对操作数据存储的SQL查询指定哪些数据将填充新快照来创建快照。初始快照始终创建为操作数据的真实和可重用副本，使数据的特定状态不可变。因此，初始快照作为后续数据整理的起点，但它始终允许回溯到创建初始快照时原始数据的确切状态。

由于已创建的快照无法更改，因此在开始数据整理之前，必须“准备”快照。准备好的快照的数据可以进行协作修改，为模型训练做准备，特别是为数据管理做准备。此外，快照通过将每个操作作为元数据记录在DB4AI系统目录中，自动跟踪所有的更改，为数据提供完整的集成历史。

快照准备完成后，可以发布快照。发布的快照是不可变的，DB4AI系统强制只有发布的快照才能用于模型训练。保证训练任务

存档过时的快照以用于文档目的。在这种状态下，数据保持不变但不能用于训练新的模型。最后，清除快照，删除模式中的数据表以及视图、恢复存储空间。需要注意的是，快照管理为了实施严格的模型来源无法清除具有依赖的快照。

利用GUC参数，snapshot使用物化存储模式或者增量存储。在增量存储模式中，新快照对应的视图和数据表只保存相对父快照修改的内容，从而大大降低存储空间。

* 1. DB4AI-Query

原生AI框架深度内嵌于数据库内核中，通过查询优化和查询执行，构建包含AI算子的执行计划。计算完成后，框架的存储模块将负责保存模型相关信息。整个AI框架主题分成3部分，分别是：查询优化模块、计算执行模块和模型存储模块。

查询优化：

框架新增词法、语法规则CREATE MODEL、PREDICT BY作为AI计算入口。在查询优化中，模块负责简单的输入校验，包括：属性名合法性、算法当前是否支持、模型名称是否冲突等。校验完成后，该模块根据训练和推测任务生成对应的查询计划。

计算执行：

查询执行模块负责根据需求算法类型的不同添加相对应的AI算子到执行计划中，并执行运算其中包括数据读取和模型计算更新。各个算法之间高内聚低耦合，具有非常好的算法扩展性，对开发者之后添加算法友好。

模型存储：

当模型完成训练，执行器会把模型数据以tuple的形式传递给存储模块，最终将模型保存到系统表gs\_model\_warehouse中。

计算机生成了可选文字:
SQL ， 四 
User 一 ． 
GaussDB Kernel 
User 丨 n 而 ce 
Query Optimizer 
E2E Al Manager & Executor 
Storage Engine 
Heterogeneous Computing 

接下来我们以CREATE MODEL为例介绍用于训练模型的查询语句是如何实现的：

计算机生成了可选文字:
] 三 曰 彐 
CREATE MODEL... 以 
| 一 
三 ] 一 
， ， ， plan 
劁 operator 
P ， 0 霾 ， ， ， ， or ， ： ution, 
， ， 0 'Crute b 
Add [ 
0 ， at 
曰 
[ 
E ． uto ， S ， ， 1 
"ecutOrhnlSh 
曰 一 

第一步对Query进行词法分析、语法分析(Lex、Yacc)。通过识别模式类别和模式组合校对语句是否存在语法错误，生成分析树。

第二步 通过词法分析、语法分析(Lex、Yacc)后，数据库会对得到的每一个分析树进行语义分析和重写。在语义分析生成查询树的过程中，针对命令类型为createmodelStmt的情况，数据库首先会对算法类型进行检查判断算法属于监督学习还是非监督学习，根据这个判断结果继而进一步校验查询语句所输入的属性、超参、模型名称是否非法等。校验完成后，语义分析生成查询树，传递给数据库执行器。

第三步 在执行阶段根据算法类型的不同，执行器会添加不同的算法算子到执行计划中，将AI算子添加到扫描算子的上层。在算子执行计算的过程中，把扫描得到的数据输入到算法模型中进行计算和更新，最后根据超参设置的迭代条件结束算子执行。

第四步 计算完成后，执行器会将已训练完成的模型以元组的形式传递给存储引擎，接收到的元组转写模型结构体，经校验保存到系统表gs\_model\_warehouse中。用户可以通过查看系统表的方式查看模型的相关信息。

DB4AI作为openGauss原创的高级特性，凝结了openGauss在AI上的全新实践，通过DB4AI进一步拓展了openGauss数据库的应用领域。利用openGauss提供的开箱即用的DB4AI功能，既有效解决数据仓库、数据湖场景中数据搬迁的问题，又提升了数据迁移过程中涉及的信息安全问题。未来，结合openGauss的多模、并行计算等领先优势，必将进一步地形成统一的数据管理平台，减少数据异构、碎片化存储带来的运维、使用困难。DB4AI特性的发布，是将openGauss进一步打造成一把锋利的瑞士军刀的关键一步！